FREIE UNIVERSITÄT
BERLIN

Information Management as an Explicit Role in OSS projects: A Case Study

Christopher Oezbek, Robert Schuster, Lutz Prechelt

B-08-05
25.04.2008

FACHBEREICH MATHEMATIK UND INFORMATIK
SERIE B • INFORMATIK

Information Management as an Explicit Role in OSS projects: A Case Study

Christopher Oezbek, Robert Schuster, and Lutz Prechelt

Freie Universität Berlin, Institut für Informatik
Takustr. 9, 14195 Berlin, Germany
{oezbek, rschuste, prechelt}@inf.fu-berlin.de

Abstract. Globally distributed teams of volunteers and communication by electronic means are at the core of Open Source Software development. To help projects in managing their information, we defined a light-weight, role-based process improvement and observed its use in a longitudinal case study. Results gathered by mailing-list analysis give insights into the different types of information managed and their relative importance: While technical content such as *how-to*s and *to-do*s is most frequent, the amount of information regarding decision making is surprisingly low.

Keywords: open source, information management, process improvement

1 Introduction

Open Source development, like software development in general, is an information-intensive activity. While co-located teams in traditional software engineering can exchange information face to face, Open Source project participants cannot, as they are globally distributed, often spanning time zones. Instead, Open Source projects make use of electronic communication, typically in the form of mailing lists [1]. Unfortunately, mailing lists have a number of drawbacks regarding accessibility of information: (1) threaded discussion delocalizes information, thus making it hard for readers to locate and extract relevant information, (2) no mechanism exists providing summarization of discourse [9, 3], and (3) since all content is archived without possibility of modification, conflicting messages may exist if information changes over time.

As a partial remedy and also to achieve other benefits, we propose a new role for Open Source projects, the information manager, whose goals are lowering the entry barrier for new members, enhancing overview of the project, summarizing and communicating decisions made in the project, and creating structures for self-sustaining information management [10]. To achieve these goals, the information manager should compile, structure, maintain, and advertise a separate repository of relevant information such as a wiki. The information is either found by monitoring the mailing lists (for decisions made, recurring questions, etc.) or explicitly written as introductory material, in particular for new project members. The role can be bound to a single person or be distributed among several.

In Malhotra and Majchrzak [6], the authors note that all successful knowledge management initiatives in their study of far-flung teams in a Fortune 500 company had established the role of a knowledge manager with very similiar goals to those envisioned for the information manager role in Open Source projects: The knowledge manager (1) "ensures that valuable information is not left unrecorded in the [...] repository", (2) "help[s] organize the knowledge in a way that it would be easy to locate for future re-use", (3) "remind [...] the team of past information and help [...] them find it when needed" (p. 85).

Given such a prescriptive definition of the process improvement, we ask two research questions: (1) Is the information manager role viable in the context of Open Source thus will it be adopted by an OSS project if proposed and jump-started? Given the no-nonsense attitude prevalent in Open Source projects, we believe that such adoption would be a strong indicator for benefice of the role to the project [7]. (2) If adoption succeeds, how will the role actually be performed? While the definition of the information manager, suggests a set of common tasks and activities such as summarzing decisions, we wanted to see whether these tasks were really relevant to the project and whether and how the process improvement would be re-invented to fit the requirements of Open Source software production better.

To answer these questions, we have conducted a 22-month exploratory longitudinal study together with the Open Source project GNU Classpath. One of the authors approached the project and offered to perform the information manager role over the course of three months. Two years later we retrospectively analysed the mailing-list to see whether and how information management had been sustained.

We will now shortly describe how we introduced information management in the GNU Classpath project and how we analyzed the long-term effects this had on the project. We then report on the results of this analysis.

2 Introducing Information Management

GNU Classpath was founded in 1998 with the aim to write a Free Software/Open Source version of the Java class libraries, which are needed to run software written in Java. This was prompted by Sun Microsystems' (now past) policy to keep their implementation free of charge but under a restrictive license, causing what Richard Stallman has called "the Java trap": OSS written in Java could not be run on an entirely free system [11]. As GNU Classpath matured progressively, the attitude of Sun changed and a release of the Java class libraries under an Open Source license has now taken place in May 2007.[1]

Information Management was introduced to the project by Robert Schuster after some time of involvement with the project. He contacted the maintainer

[1] Announcement of OpenJDK release `http://mail.openjdk.java.net/pipermail/announce/2007-May.txt`

with the proposal to establish information management based on a WikiWiki system as the knowledge repository. The maintainer welcomed the idea and championed the idea with the rest of the project. After creating some initial content, the wiki was publically announced to the project on January 16, 2005. The information manager then primed the wiki with relevant information over the course of three months, during which he also encouraged its use and frequently referred project members to information in the wiki. After this period, our official involvement with GNU Classpath ended.

3 Evaluating Information Management

We extracted all 8588 e-mails sent to the developer mailing list of GNU Classpath since the beginning of its archival in March 2002 up to September 2006. To identify messages having to do with information management, we used a full text search for the terms *wiki*, *mediation*, and *mediator* (mediator was the term we chose for the information manager when starting the case study). In particular the term *wiki* was so closely connected with the information management effort that we believe this query has a high-enough recall to ensure representativeness and validity. The search returned a total of 280 messages.

These 280 e-mails were then read and categorized manually. 175 messages were found to actually relate to the information management effort, while the other 105 pointed to other Wikis (such as Wikipedia) or had different topics and contained a search term only in text cited from a previous message. The latter messages were not considered further. We then categorized the 175 relevant messages into these topic classes:

- Meta-discussion, i.e. discussion about information management (36 messages).
- Actual information management acts (133 messages).
- A discussion about using wikis in Classpath that predates the information mangement effort (4 messages).
- E-mails related to our study of information management(2 messages).

Looking at the temporal distribution of messages, we see that meta-discussion was more frequent in the beginning of the information management effort and then decreased, while actual information management picked up slowly and was well established after about one year, see Figure 1. We will now discuss both of these types of messages in detail.

3.1 Meta Discussion

In the beginning of the introduction the discussion was dominated by messages about information management such as how to implement the information management role or which kind of information should be stored in the wiki. We recognized six recurring types of content.

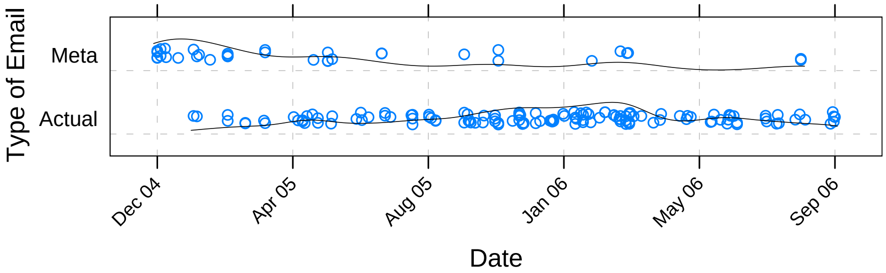

Fig. 1. Messages plotted over time, distinguishing messages in which information management is discussed (meta) and actually performed (actual).

6 messages *announce*d the availability of a new use or information type, such as the announcement to include *to-do*s and to create a page for organizing real-life meetings. Interestingly, such new areas were never proposed before being announced — all announcement was after the fact.

13 messages were *discuss*ions on information management. Examples include which kind of time format to use in the wiki, or whether it is an advantage for the information management effort that participants who have looked at the source code of the Sun JDK may not write code for GNU Classpath but must find other ways to contribute.[2]

On top of these, two specific types of opinion offered were *praise* (2 messages) and *critic*ism (6 messages), which were voiced regarding the information management effort as a whole or regarding the behavior of individuals in this context. For instance, criticism occured when it became apparent that *to-do* information was now located in three places (static web site, *to-do*-tracker, wiki); the author of the post suggested moving all such information to the wiki and removing the other two. Other criticism suggested not to use the wiki for discussions.

5 messages explicitly *thank*ed the information managers in the project for their contributions. 4 messages concerned overtly *technical* issues related to the wiki server (after disk space problems) or the wiki software.

Figure 2 shows the temporal distribution of each meta discussion contribution type. Except for criticism, all kinds occured early. Criticism started only when the effort matured. Since criticsm messages may provide important information whether information management is well received, we sketch and discuss each of them.

1. “We now have *to-do* information in three different places; that’s bad. Let’s keep it all in the wiki only.” The project followed this suggestion.

[2] Developers who have been “tainted” by looking at code written by Sun are excluded from participating in GNU Classpath to safeguard against copyright violations.

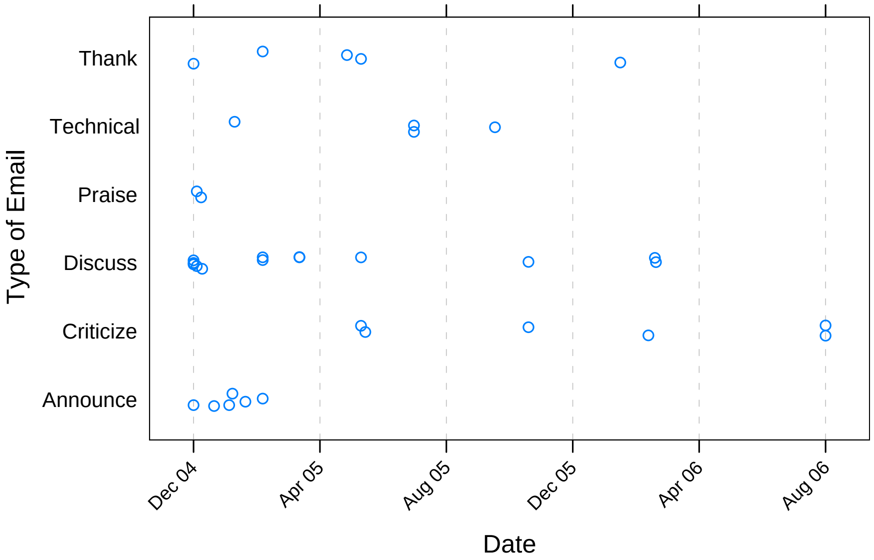

Fig. 2. Distribution of meta discussion types over time. Each circle represents one message.

2. “Do not abuse the wiki for discussions.” This is in line with the well-known relative weakness of wikis for discussion [5] compared to more structured forums or mailing-lists. Purely wiki-based projects like Wikipedia have created explicit discussion pages attached to each wiki page to facilitate such discourse [8, 12], others have established guidelines for how to discuss an issue in *thread mode* in contrast to presenting information in *document mode* [4]. Technical solutions have also been proposed such as qwikWeb, which turns mailing-list discussions into wiki pages to gain the benefits of both mediums [2].
3. “The important page !!! is difficult to navigate to.” This points to the need of wikis to be ‘gardened’ by ‘wiki gnomes’, i.e. maintained by dedicated members for quality and efficient access. The role of the information manager was designed with this issue in mind, and we believe that the criticsm strengthens our case that explicitely defined information management is quite different from just “having a wiki”.
4. “I do not like wikis.”, which is not further elaborated upon by the author.
5. “We should not force-and-record a decision on <issue-so-and-so>, but rather leave it to individual common sense.” This criticism is most interesting with respect to the dynamics of Open Source projects. As we will see in the next section, it points to the prevalence of implicit decision making.

To sum up these criticisms, they appear to indicate that the members of the project appreciate the information management platform and role, but that careful shaping of the role is required.

3.2 Managing Information

After looking at the meta-discussion of information management, a next step was to explore in more detail how the innovation was actually used. We found 133 messages that directly relate to creating or using information in the wiki and categorized them with respect to *information type* and *action type*. There are six action types:

- *Create* (10 percent or 13 out of 133) denotes a message announcing new information items in the wiki.
- *Add* (9 percent or 12 out of 133) is used when one or several items in the wiki have been updated or added to.
- *Synchronize* (14 percent or 19 of 133) indicates messages related to moving information from mailing list to wiki, and contains questions for clarification (10), answers to such questions without updating the wiki (3), requests for putting something into the wiki (5), and answers indicating that a wiki entry is being worked on already (1).
- *Use* (17 percent or 23 out of 133) is for messages that contain information that was explicitly taken from the wiki.
- *Refer* (30 percent or 39 out of 133) messages suggest to retrieve information from the wiki. They either provide a direct link or mention the existence of a topic or starting point.
- *Answer* (20 precent or 27 out of 133) messages are replies that acknowledge the receipt of information, thank the author, or veer off-topic, etc. These 27 e-mails do not contribute new information nor make use of existing content and were thus excluded from further discussion, leaving 106 remaining messages.

In total there are 62 messages that consume information from the wiki (*use* and *refer*) and 44 messages that generate information for the wiki (*create*, *add* and *synchronize*), thus a ratio of roughly 2:3 of messages concerning generated and consumed information, respectively.

When looking at the temporal distribution of messages with respect to their action category (Figure 3), we can conclude

- the effort to establish information management is almost immediately useful to the project, as referal and usage follows creation almost immediately,
- usage, referal and synchronisation arise continuously, whereas
- creating new topics decreases over time while updating becomes more important.

Although these data suggest that the amount of information in the wiki stops growing after some time, we cannot be sure this is really the case. It is easily conceivable that future project events will trigger a new wave of additions.

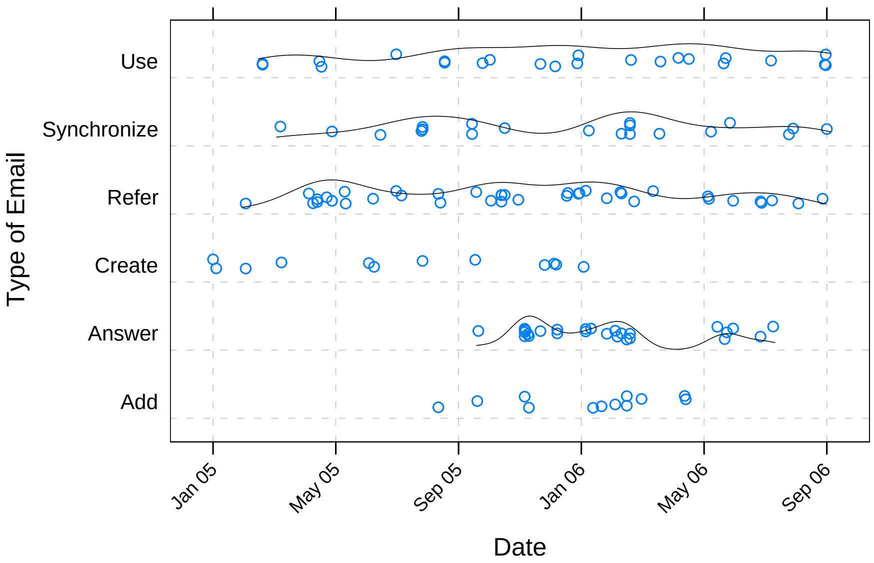

Fig. 3. Distribution of information management action types over time. Each circle represents one message. The lines are gaussian-kernel density estimates.

After analyzing which kind of action was indicated by each message, we studied the information type of the messages (excluding the 27 *answers*). We found that the project had invented additional uses for the wiki that went beyond those envisioned in the original information management idea.

Only two messages (1.9 percent of 106) were found to relate to *decisions* that the project deemed important enough to transfer into the wiki; this category was of much less importance than we had anticipated. We identified two reasons: First, many would-be decisions were actually recorded in *how-to* documents and messages (often reflecting the existence of a canonical view). Second, as reflected in the respective criticism mentioned before, the project's mindset prefers leaving room for individual, common-sense decisions.

We already mentioned *to-do*s in the wiki. A total of 10 messages (9 percent of 105) belong to this category and relate to individuals announcing their current *to-do*-status or refering others to pick open tasks from the wiki.

*How-to*s, i.e. descriptions of how to install the project, compile it, submit contributions, etc. are the most frequent information type (40 percent or 42 out of 106). They are not only refered to by experienced developers in their communication with new members, but are also repeatedly used even by those developers who wrote them, which explains their popularity. An example is the "How to contribute to GNU Classpath" document explaining how to employ the tools such as the Eclipse IDE, Mauve test-suite, Open Source virtual machine

Cacao, etc. The *how-to* started as a white paper, turned into a wiki page, and then rapidly expanded to become more user-friendly and complete.

The second-most important information type are instructions for new members (*for-newbies*) of the project, especially the formal process for gaining CVS commit rights in GNU Classpath. It caused a total of 20 messages (19 percent of 106) to be sent in this category.

As one of the unexpected information types, we saw the use of the wiki as a platform to *organize* real-world meetings of the project (8.5 percent or 9 out of 106). The two notable use cases were the collection of phone numbers prior to an Open Source conference and the voting for the after-conference pub to visit.

Representation issues rank third (15 percent or 16 out of 106). About three months after the wiki started, some project members suddenly began to create a gallery of project successes: applications that are now supported to run using the Classpath Java libraries. A flurry of activity occured in this category about six months later after significant progress had been made on GNU Classpath.

Lastly, 7 messages (6.6 percent of 106) sent by the maintainer of the project to announce new releases referred to the information management activity in some *general* way. The temporal distribution of messages from each content type is shown in Figure 4.

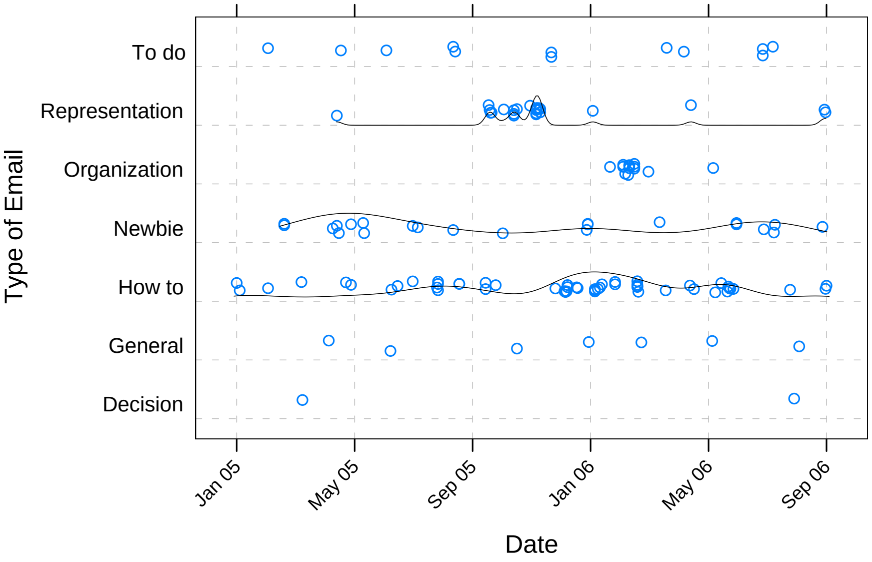

Fig. 4. Distribution of information management information types over time. Each circle represents one message. The lines are gaussian-kernel density estimates.

3.3 Threats to validity

It is difficult to judge the external validity (generalizability) of the study, as it was the first of its kind. Further work will be required to understand to what degree other circumstances than this specific project will evoke similar ease-of-adoption, action type patterns, and information type distributions. We would expect differences if the maintainer is less committed to the effort and does not champion the introduction.

The internal validity is threatened by three issues. First, we may have missed some relevant messages on the mailing list, which may have distorted the results. While a few missed messages are likely, we are confident that the distortion is negligible. Second, the categories we chose may be inappropriate or some categorizations wrong. As both threats are rather straightforward, we would not expect problems in this regard. Third, other materials than mailing list messages (in particular the wiki pages) were used only for understanding the information in the messages and not evaluated separately. Adding such analysis might reveal additional details but is unlikely to change the overall results.

4 Conclusion

We have suggested a process improvement for OSS projects in the form of a new role: the information manager.

We have evaluated the suggestion by means of a 22-month case study in the context of the GNU Classpath Java library replacement project. The results suggest that the role was beneficial to this project, as it was adopted quickly and then self-sustained without further intervention.

We found that information management was mainly used for maintaining information on how to solve technical problems, what new members should do to participate, to organize real-world aspects of the project, manage *to-do*s, and represent the project to the rest of the world. Contrary to our expectations, we have hardly seen it being used for tracking decision-making in the project and explained this with a preference for individual common-sense decision making and recasting decision processes as a search for a technical solution.

Meta discussion on information management as a new process improvement was frequent in the beginning of the project, but largely died down after a few months.

The sequence of events suggests that it is valuable to have support from core members of a project during the introduction phase.

Acknowledgments

We would like to thank the members of GNU Classpath for working together with us to implement this process improvement in their project.

References

1. Davor Cubranic and Kellogg S. Booth. Coordinating Open-Source Software development. In *WETICE '99: Proceedings of the 8th Workshop on Enabling Technologies on Infrastructure for Collaborative Enterprises*, pages 61–68, Washington, DC, USA, 1999. IEEE Computer Society.
2. Kouichirou Eto, Satoru Takabayashi, and Toshiyuki Masui. qwikWeb: Integrating mailing list and WikiWikiWeb for group communication. In *WikiSym '05: Proceedings of the 2005 International Symposium on Wikis*, pages 17–23, New York, NY, USA, 2005. ACM Press.
3. James Hewitt. Beyond threaded discourse. *International Journal of Educational Telecommunications*, 7(3):207–221, 2001.
4. Ben Kovitz. How to converse deeply on a Wiki. Why Clublet, February 2001. `http://clublet.com/c/c/why?HowToConverseDeeplyOnAWiki` visited 2007-06-21.
5. Kevin Makice. Politicwiki: exploring communal politics. In *WikiSym '06: Proceedings of the 2006 international symposium on Wikis*, pages 105–118, New York, NY, USA, 2006. ACM Press.
6. Arvind Malhotra and Ann Majchrzak. Enabling knowledge creation in far-flung teams: best practices for IT support and knowledge sharing. *Journal of Knowledge Management*, 8:75–88, April 2004.
7. Christopher Oezbek and Lutz Prechelt. On understanding how to introduce an innovation to an Open Source project. In *Proceedings of the 29th International Conference on Software Engineering Workshops (ICSEW '07)*, Washington, DC, USA, 2007. IEEE Computer Society.
8. Christian Pentzold and Sebastian Seidenglanz. Foucault@Wiki: First steps towards a conceptual framework for the analysis of Wiki discourses. In *WikiSym '06: Proceedings of the 2006 International Symposium on Wikis*, pages 59–68, New York, NY, USA, 2006. ACM Press.
9. Paul Resnick, Derek Hansen, John Riedl, Loren Terveen, and Mark Ackerman. Beyond threaded conversation. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 2138–2139, New York, NY, USA, 2005. ACM Press.
10. Robert Schuster. Effizienzsteigerung freier Softwareprojekte durch Informationsmanagement. Studienarbeit, Freie Universität Berlin, September 2005.
11. Richard M. Stallman. Free but shackled - the Java trap, April 2004. `http://www.gnu.org/philosophy/java-trap.html`, visited 2007-06-19.
12. B. Stvilia, M. B. Twidale, L. Gasser, and L. C. Smith. Information quality discussions in Wikipedia. Technical Report ISRN UIUCLIS–2005/2+CSCW., University of Illinois at Urbana-Champaign, 2005.